

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

## **1. Введение**

В последние десятилетия интернет стал неотъемлемой частью жизни миллиардов людей по всему миру. Мы ежедневно используем его для работы, общения, покупок и других задач, что привело к появлению новых угроз безопасности. Одной из самых опасных является фишинг — тип кибератаки, при которой злоумышленники маскируются под надежные источники, чтобы обманом получить конфиденциальные данные, такие как пароли и данные банковских счетов. Фишинг часто реализуется через фальшивые веб-сайты или электронные письма, которые кажутся легитимными. Неопытные пользователи могут стать жертвами таких атак, что приводит к кражам данных и финансовым потерям.

Несмотря на существующие меры безопасности, включая двухфакторную аутентификацию, фишинговые атаки продолжают оставаться эффективными, затрагивая миллионы пользователей ежедневно. Это подчеркивает необходимость разработки новых решений для их защиты. Одним из таких решений может стать антифишинговый плагин для браузера, способный анализировать веб-страницы, выявлять потенциально опасные ресурсы и блокировать их. Это существенно повысит уровень безопасности пользователей в сети.

Разработка антифишингового плагина требует создания алгоритмов, способных точно распознавать фишинговые сайты, минимизируя ложные срабатывания. Также важно обеспечить совместимость плагина с различными браузерами и операционными системами и сделать его простым в установке и использовании. Антифишинговый плагин может

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

значительно снизить количество успешных фишинговых атак, защищая пользователей от кражи данных. Этот проект не только предлагает техническое решение, но и способствует повышению осведомленности пользователей о киберугрозах. Основная цель проекта — создание эффективного инструмента для защиты от фишинга, который поможет сделать интернет безопаснее.

## **2. Условия**

### **1. Совместимость с Google Chrome:**

- плагин должен быть разработан с учетом требований Google Chrome, используя Chrome Extension API, и совместим с последними стабильными версиями браузера.

### **2. Распознавание фишинговых сайтов:**

- плагин должен в реальном времени анализировать URL и содержимое веб-страниц для определения потенциально фишинговых сайтов, учитывая методы маскировки (подмена доменов, похожие символы, фишинговые формы).

### **3. Интеграция с браузером:**

- плагин должен интегрироваться с интерфейсом Google Chrome, предоставляя пользователю интуитивно понятный интерфейс для управления. Интерфейс должен включать иконку в панели инструментов и отображать уведомления при обнаружении фишинговых сайтов.

### **4. Эффективность и точность:**

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

- плагин должен обеспечивать высокую точность распознавания фишинговых сайтов с минимальными ложными срабатываниями, не влияя на производительность браузера.

5. Безопасность и конфиденциальность:

- плагин должен соответствовать строгим стандартам безопасности, предотвращать утечки данных и быть прозрачным в отношении сбора и использования данных, позволяя пользователям управлять настройками конфиденциальности.

6. Удобство использования:

- плагин должен быть простым в установке и использовании, с интуитивно понятным интерфейсом, содержащим четкие инструкции и подсказки. Включение раздела справки или FAQ будет преимуществом.

7. Тестирование и качество:

- плагин должен пройти тщательное тестирование на всех этапах разработки, включая функциональное и производительное тестирование, с особым вниманием к совместимости с различными версиями Google Chrome и другими расширениями.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

### **3. Техническое задание**

*Функциональное задание:*

Функциональное задание описывает основные и дополнительные функции, которые должен выполнять разрабатываемый антифишинговый плагин для браузера Google Chrome. Задание также включает требования к пользовательскому интерфейсу, чтобы обеспечить удобство и эффективность использования плагина.

*Обязательная функциональность:*

#### **1. Распознавание фишинговых сайтов.**

- Плагин должен проверять URL и содержимое загружаемых страниц на наличие признаков фишинга, используя заранее определенные списки и правила.
- Источники данных для проверки: плагин использует обновляемые списки URL, классифицированные как фишинговые, например: OpenPhish, PhishTank или Google Safe Browsing API.
- Правила и эвристики: в дополнение к спискам плагин должен иметь набор правил и шаблонов для выявления подозрительных признаков, таких как структура URL, необычные элементы HTML или JavaScript, характерные для фишинговых сайтов.

#### **2. Блокировка фишинговых сайтов.**

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

- Реализация блокировки до полной загрузки страницы: при обнаружении фишинговой угрозы плагин должен предотвращать загрузку содержимого страницы, чтобы снизить риск взаимодействия пользователя с вредоносным сайтом. Блокировка должна выполняться с использованием Chrome Extension API, особенно API webRequest, чтобы вмешиваться в процесс загрузки до его завершения.
- Оповещение пользователя: в случае блокировки пользователю должно выводиться уведомление о потенциальной угрозе с объяснением причин блокировки.

### 3. Управление белыми списками.

- Интерфейс управления белыми списками: плагин должен предоставлять возможность пользователю добавлять домены в белый список. URL из белого списка не будут проходить проверку и блокировку.
- Проверка белых списков: при обращении к сайту плагин проверяет, находится ли домен в белом списке, и при необходимости исключает его из блокировки. Белый список должен храниться локально для быстрого доступа и минимизации задержек.

### 4. История блокировок.

- Плагин должен сохранять историю заблокированных сайтов, включая URL, дату блокировки и причину блокировки.
- Пользователь должен иметь возможность просматривать и фильтровать историю блокировок.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

5. Настройки конфиденциальности.

- Плагин должен позволять пользователю управлять сбором статистики, отправкой отчетов, и возможностью удаления истории блокировок.

*Дополнительная функциональность:*

1. Настраиваемые уровни фильтрации.

- Пользователь должен иметь возможность настраивать уровень строгости фильтрации с помощью параметров, таких как ключевые слова, черные списки доменов и другие правила.

2. Выгрузка отчетов.

- Плагин должен предоставлять возможность выгрузки отчетов о проведенных блокировках в формате CSV или JSON.

3. Интеграция с внешними сервисами.

- Возможность интеграции плагина с внешними сервисами для проверки URL на предмет фишинговых угроз.

*Требования к пользовательскому интерфейсу:*

1. Настраиваемые уровни фильтрации.

- Пользователь должен иметь возможность настраивать уровень строгости фильтрации с помощью параметров, таких как ключевые слова, черные списки доменов и другие правила.

2. Выгрузка отчетов.

- Плагин должен предоставлять возможность выгрузки отчетов о проведенных блокировках в формате CSV или JSON.

3. Интеграция с внешними сервисами.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

- Поддерживаемые внешние сервисы: интеграция с API популярных антифишинговых сервисов, таких как Google Safe Browsing, PhishTank и OpenPhish, позволяет использовать актуальные данные для обнаружения угроз. Эти сервисы предоставляют обновляемые списки фишинговых URL и улучшают точность проверки.
- Архитектурные особенности.
  - Асинхронные запросы и кеширование: чтобы минимизировать задержки, плагин должен использовать асинхронные запросы и кратковременное кеширование результатов проверок.
  - Очередь запросов и лимиты API: для снижения нагрузки и предотвращения блокировок из-за лимитов API важно настроить очередь запросов и тайм-ауты, что обеспечит устойчивую работу.
- Функциональность.
  - Обновляемые данные: внешние сервисы обеспечивают актуальные данные для предотвращения фишинга, однако могут вызывать сетевые задержки.
  - Резервные проверки: в случае недоступности внешних API плагин может использовать локальные списки, чтобы не терять функциональность.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

#### **4. Рекомендации к выполнению**

##### 1. Исследование и анализ угроз.

- Анализ решений: изучите текущие антифишинговые плагины и методы, определите их сильные и слабые стороны для разработки более эффективного решения.
- Обновление базы данных: регулярно обновляйте базы данных фишинговых сайтов и черные списки для актуальности.

##### 2. Проектирование и архитектура.

- Модульная архитектура: используйте модульный подход для упрощения обслуживания и добавления новых функций.
- Пользовательский интерфейс: разработайте прототипы интерфейса, протестируйте их с пользователями, обеспечивая удобство и интуитивность. Используйте принципы UX/UI.

##### 3. Разработка и тестирование.

- Кодирование: следуйте лучшим практикам программирования и соблюдайте стиль кода для улучшения читаемости и поддержки.
- Тестирование: проводите функциональное и нефункциональное тестирование, включая тестирование производительности и безопасности. Используйте автоматизированные тесты.
- Совместимость: убедитесь, что плагин корректно работает с разными версиями Google Chrome и на различных операционных системах.

##### 4. Безопасность и конфиденциальность.



**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

- Безопасность данных: применяйте надежные методы защиты данных и шифрования для предотвращения утечек.

5. Интерфейс и удобство использования.

- Простота и ясность: сделайте интерфейс простым и понятным, чтобы пользователь легко находил и использовал все функции плагина.

**5. Требования к документации**

- Титульный лист (с указанием названия кейса и перечислением членов команды).
- Обоснование выбора языка программирования и используемых программных средств.
- Структурная и функциональная схемы программного продукта.
- Блок-схема работы основного алгоритма.
- Описание особенностей и аргументация выбранного типа СУБД.
- Схема базы данных.
- Программный код (ссылка на репозиторий), файл README должен включать:
  - краткое описание проекта;
  - инструкцию по установке/развертыванию;
  - ссылку на видеоролик.

**6. Требования к видеоролику**

- Видеоролик должен демонстрировать функционирование разработанного программного продукта в соответствии с регламентом испытаний.
- На видео или записи экрана необходимо продемонстрировать выполнение каждого испытания, описанного в регламенте, в соответствии с условиями.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**

**Профиль «Информационные технологии»**

**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

- Видео должно однозначно подтверждать авторство участников (во время записи ролика необходимо четко произнести название команды, ФИО участников, номер школы, ФИО руководителя).
- Видеоролик необходимо разместить на стороннем видеохостинге («ВКонтакте», Rutube и др.)

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

## **7. Регламент испытаний**

Этапы проведения испытаний

### 1.1. Подготовка к испытаниям.

- Установка плагина: установить антифишинговый плагин на последнюю стабильную версию браузера Google Chrome на нескольких тестовых устройствах с различными операционными системами (Windows, macOS, Linux).
- Настройка окружения: обеспечить доступ к интернету и подготовить тестовые аккаунты, если необходимо. Создать список тестовых URL, включающий как известные фишинговые сайты, так и легитимные сайты.
- Инструменты мониторинга: настроить инструменты для мониторинга производительности и использования ресурсов плагином (например, Chrome Developer Tools, Task Manager).

### 2. Функциональное тестирование.

#### 2.1. Распознавание и блокировка фишинговых сайтов.

- Тест 1: Блокировка известных фишинговых сайтов.
  - Откройте в браузере сайт, известный как фишинговый, и убедитесь, что плагин успешно распознает его и блокирует доступ.
  - Зафиксируйте уведомление плагина, которое должно содержать информацию о причине блокировки и опции для пользователя (например, «Подробнее», «Исключить из блокировки»).
- Тест 2: Распознавание нового фишингового сайта.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

- Создайте или выберите новый фишинговый сайт, который еще не был добавлен в черные списки, и откройте его в браузере.
- Проверьте, как плагин распознает и блокирует этот сайт. Зафиксируйте результат.

## 2.2. Работа с белым списком сайтов.

- Тест 3: Добавление сайта в белый список.
  - Добавьте безопасный сайт, который плагин ошибочно определяет как фишинговый, в белый список.
  - Откройте сайт снова и убедитесь, что плагин не блокирует доступ к этому сайту.
- Тест 4: Удаление сайта из белого списка.
  - Удалите сайт из белого списка.
  - Откройте сайт снова и проверьте, что плагин снова проверяет сайт и блокирует его, если он подозрительный.

## 2.3. Тестирование производительности.

- Тест 5: Влияние плагина на время загрузки страниц.
  - Замерьте время загрузки нескольких сайтов с включенным и отключенным плагином.
  - Сравните результаты и оцените, насколько плагин влияет на производительность браузера.

## 3. Тестирование стабильности и надежности.

### 3.1. Поведение после сбоя.

- Тест 8: Восстановление после сбоя.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

- Принудительно завершите работу браузера и затем откройте его снова.
- Проверьте, сохраняются ли настройки плагина, и продолжает ли он корректно выполнять свои функции после перезапуска браузера.

#### 4. Тестирование пользовательского интерфейса.

##### 4.1. Интерфейс и удобство использования.

- Тест 9: Оценка удобства интерфейса.
  - Проанализируйте, насколько интуитивно понятен и удобен интерфейс плагина для пользователя.
  - Проверьте доступность всех основных функций, таких как управление белым списком, настройка уровней защиты и просмотр истории блокировок.

##### 4.2. Работа уведомлений.

- Типы уведомлений:
  - Оповещение о блокировке фишингового сайта: при попытке пользователя открыть фишинговый сайт плагин блокирует страницу и выводит уведомление.
  - Информация о причине блокировки: уведомление должно содержать причину блокировки (например, ссылка определена как фишинговая по данным конкретного сервиса или правилам плагина).
- Тест 10: Проверка системы уведомлений.
  - Создайте условия для генерации уведомлений плагина (например, при попытке посетить фишинговый сайт).

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

- Убедитесь, что уведомления отображаются своевременно, корректно и содержат необходимую информацию для пользователя.

## 8. Методические материалы

1. [Python](#) – официальный сайт языка программирования Python. Необходим для разработки вспомогательных инструментов и скриптов, а также для тестирования.
2. [PyCharm](#) – интегрированная среда разработки (IDE) для Python. Предоставляет мощные инструменты для редактирования, отладки и тестирования кода.
3. [Selenium WebDriver](#) – инструмент для автоматизации веб-браузеров, который может быть использован для тестирования работы плагина в браузере Google Chrome.
4. **Flask** – легковесный веб-фреймворк для Python, который может быть использован для разработки серверной части, если требуется взаимодействие плагина с сервером.
5. [pip](#) – официальный менеджер пакетов Python, который используется для установки и управления библиотеками и зависимостями проекта.
6. [Git](#) – система контроля версий, которая позволяет отслеживать изменения в коде и управлять проектом совместно с командой.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

7. [GitHub](#) – платформа для хостинга Git-репозитория, предоставляющая возможности для управления проектами, совместной разработки и публикации исходного кода.
  
8. [Jupyter Notebook](#) – интерактивная среда для разработки и выполнения кода Python, особенно удобная для прототипирования и анализа данных.
  
9. [Sphinx](#) – инструмент для создания документации из исходного кода Python. Поддерживает генерацию документации в различных форматах.
  
10. [pytest](#) – фреймворк для тестирования Python, который поддерживает как модульное, так и интеграционное тестирование.
  
11. Chrome Developer Tools – встроенные в Google Chrome инструменты разработчика, которые помогают отлаживать и тестировать расширения и веб-приложения.
  
12. [Anaconda](#) – дистрибутив Python, который включает в себя множество научных библиотек и инструментов для анализа данных и машинного обучения, что может быть полезно при разработке сложных алгоритмов для плагина.